JUNE '84

I-M 1 IN A MILLION CLUB

NATIONAL NEWSLETTER

HEXMART is dedicated to providing the ATE computer owner with a collection of quality utility programs. We are not a game company, we prefer to think of ourselves as a company that provides the "tools" to assist you with your programming needs.

SOFTWARE

DISK LOADER 1 The BASIC line renumbering program.
W14.95 on cassette - loads in 8K, runs using no program RAM

When you enter programs yourself into a computer? You need a text line subroutine but lose space for only five lines. Don't worry, HEXMART is here! Renumber your entire program from the beginning OR renumber starting from ANY line. You specify the starting point, load your program and CALL HEXMART. It does the rest, including changes to all line references. It's loaded with error checking which makes it easy to use.

DISK EDIT 1 A machine language line editing routine.
W14.95 on cassette - loads in 6K, requires less than 1/2 K

If you have ever tried to edit a long string of characters in BASIC, then you know how slow ATE can be. No more! SUPER EDIT is a mark-up between software routine which can be included in your programs to speed editing operations to over 100 times. A fantastic reduction of time in all kinds of edits. You can include SUPER EDIT in your customer programs it is supplied with complete instructions. Allows your program to be loaded "behind" the machine code, or you can write a new program behind SUPER EDIT, a truly powerful feature.

TAPE ASSEMBLER 1 A cassette based assembler for ATE's Hardata LABS.
W14.95 on cassette - loads in 8K, expandable to 16K

No, you don't need a disk system to write Assembly language programs. Hexmart's TAPE ASSEMBLER allows you to write these programs using 6800 code. You run and execute the following! It has programs - screen only and line printer versions. 2) Instructions - including how to use and incorporate Assembly code to any ATE. 3) 6800 Assembly language work sheets, and 4) A brief introduction to Hexmart's language programming. Learn to make the ATE fly!

DISK FILE 1 A master base directory filing system.
W13.95 on cassette - loads in 8K, transferable to diskette.

This is a complete "house file" of all diskette files. Keep and use any program that contains that needed program? DISK DIR is the answer. Creates a master file of up to 200 program files from the directories of all your diskettes. You then each disk with any three character code and DISK DIR automatically stores all program names on the disk, store them into alphabetical order, and stores them in the master file. Directory also makes the master file to be updated on your collection of programs change.

DISK HELP / UTILITY 1 A full purpose, on disk utility set of tools.
W13.95 on cassette - for ALL serious disk users.

DISK HELP - Recovers lost data! Scan a lost diskette and restore it on an AVE disk, display it on the screen, place it in RAM, modify it, or print a list of each file. Automatic diagnosis allows work on diskettes or separate backups.
DISK UTIL - Allows you to read or turn any track or sector from an AVE disk, display it on the screen, place it in RAM, modify it, or print it. Also works with diagnostics or separate backups. DISKETTE FIX utility tools allow the file routines including a screen printout, disk dump of entire programs for editing, single program routines, and print file and fixes beyond RVE. These UTILities tools are fully compatible with AVE's RVE. However when used with DISK HELP, you get an additional 12K of storage. These two programs are a must for any serious disk user.

DISKFILE 2 A data base management system.
W19.95 on cassette - loads in 16K, Requires usable DISK drive.

The first fully flexible data base management system for the AVE has arrived! No longer write your own custom programs for sorting lists, output invoices, mailing labels, parts inventory, etc. SUPERFILE reads these and many more fully filled data base!! Electrically, allows the fields and record, up to 250 records per file. Field titles and field sizes are user specified. Each on any fit up to 8, 16, or 24K systems. Extremely useful.

DISK BACKUP 1 New from HEXMART, a disk file backup program.
W12.95 on cassette - loads in 8K, transferable to disk.

Not to be confused with other backup programs! DISK BACKUP copies any type of AVE file from one disk to another. Copies files and record in sequence, directory to another disk. Copy single files or complete disks. Supports fast copy routines and file verification on all 8, 16, or 24K systems. Extremely useful for backups by data files.

GENERAL NEWS

TERM-1 & STORE DISPLAY PROGRAMS

DOUG SMITH-Fairfax, VA has been generous in submitting two original APF programs to the program library. These programs, as well as all others in the club library are FREE to all members as explained in the MARCH issue. The requirements for obtaining these programs are also stated in that issue. Activity in the library has been very LOW thus far, so we urge you to ACTIVATE IT!

TERM-1 is the software required to operate a modem with your IM-1 providing your computer is equipped with either a SI-232 or DC-232 serial interface. This program operates upon case characters only to the LINE, and any lower case characters coming in from the line will be converted to upper case when displayed on the screen. The program is a very basic terminal emulator program but does the job adequately and will allow you to access other systems via the telephone. The library designation is DBTERM1.

The STORE DISPLAY program was used by retailers to allow customers to see the IM-1's graphic power and sound quality. The continuous complex and colorful HI RES graphic screens are extremely well done and are enjoyable to watch. If you want to see a good example of what the IM-1 can do graphically and experience a MORAL BOOSTER at the same time, be sure to include this program in your library selection. Program designation DBSTOR8.

Thanks again Doug for coming through for the club!

VINCENT JOHNSON-APF RESEARCH

To insure proper credit is given to the writers of programs appearing in the library, VINCENT JOHNSON has been instrumental in providing us with the following information:

DBMOUSE was written by ROCCO REBUSTATIN and APF's "YOUR LIFE WILL NEVER BE THE SAME" newsletter in Vol 1, NO.5, page 8.

DBHUMPUS was written by MIKE RUSSELL, DBALSORT and DBINSORT were both written by CHUCK CLANCY.

There is some doubt about who the writer of DBDATAZAP could be. If anyone knows, please let us know.

Thank you Vincent for taking the time to find this information!

SCREENWRITER II NOW AVAILABLE

The most recent donation to the program library is one from KEITH PHILLIPS-San Jose, CA.

This program is in the form of a SCREEN PRINTER program that allows for hard copy reproduction of characters as they appear on the screen.

The program exhibits full screen editing at all times with CHANGE/DELETE/INSERT functions included. Both joystick/pad and keyboard are used. Complete step-by-step instructions come with the program and it is available from the library as designation DBSCRWRII.

Many hours have gone into this complex program and it is written in BASIC with machine language routines incorporated for speed. MANY THANKS KEITH.

NEW PROGRAM COMING SOON

JIM CLAYSLISTER has sent us a sneak pre-view of his newest creation called RECORD WRITER.

The program menu appears to be very similar to Jim's other word processing/business management programs, thus making it very easy to use if you are already familiar with the menu contained in the others. Even if you're not familiar with the other programs, RECORD WRITER is very easy to operate.

The program permits storage of user-defined records (10 lines each) and also contains SORTING, EDITING, and TAPE OR DISK SAVING features.

RECORD WRITER will be reviewed next month in detail along with several other excellent programs written for the IM-1. Thanks to the efforts of those who contribute herein, we now have HIGH QUALITY programs available to us. The HIGH STANDARDS you have set are most sincerely appreciated!

ART CONTEST JUDGING IN PROGRESS--RESULTS SOON

If this issue arrives at your house after JUNE 10, the contest is officially closed. Copies of all entries are being forwarded to the judges and we are awaiting the scores. The winner will be announced in the JULY issue and the GRAND PRIZE AWARD will be given.

If you receive this issue before JUNE 10, there's still time to get your 10 RES FRONT SCREEN in before the deadline. We'll be looking forward to it. HURRY!!!

HELPFUL HINTS

This category will be EXTREMELY LENGTHY this month in order to publish as much information as we can out of the large amount we have received. Because of this, some of the regular categories will not appear in this issue in order that we may bring you the following SPECIAL INFORMATION. We will return to our REGULAR SCHEDULE NEXT MONTH. I think I've been watching too many SPECIALS! In any event, the first HELPFUL HINT comes to us from BILL LUNQUIST as a result of a question submitted by BILL BOWMAN. Mr. Bowman asked, "How does one use a STRING VARIABLE to name a disc file using DIRECT ACCESS? Here's an example of the program I'm using when trying to generate a DIRECT FILE using the string variable was:

```
100INMAS(8)
200INPUT"NAME",NAS
300CLOSE1:OPEN3NAS
310POS1=17012
320PRINT#1:A,B,C,etc.
330NEXT
```

The above is fine in SEQUENTIAL, but how do I let the computer know that I WANT DIRECT? If I add, or try to add a "D" after the NAS (Line 100), the computer says "LINE 100 WHAT". If I spell out the name-no problem-but I want to be able to change it with an INPUT and still get DIRECT ACCESS. INPUTING the RECORD # works OK but NOT THE FILE NAME."

Mr. Lunquist's reply is printed here exactly as it was received and a copy of the letter was immediately sent to Mr. Bowman. A later reply indicated that the solution worked, and the problem was resolved. Our thanks to BILL LUNQUIST for his detailed explanation and examples, and to BILL BOWMAN for his question. Here is the letter from BILL LUNQUIST:

GEORGE BAKER
RED GRAFIX LIMITED
P.O. BOX 54
ARROWSMITH, ILL. 61722

Dear Mr. Baker,

I have two methods for dealing with opening a direct access file on the APF using a string variable for the file name. But first of all, there are a few things you should know about APF files. The single letter file type that is displayed when you do a DIRectory command has no bearing on how APF treats that file. That letter is generated when the file is saved or opened for the first time and never looked at again.

If you write a long sequential file and then use the same file name to write a shorter file, the disk directory does not free up the disk space that was assigned to the longer file. The space is freed if you kill the file.

If you kill a direct access file, the space it occupied on disk may or may not be made available. The people at HEXMART SOFTWARE are working on some disk utilities one of which will correctly kill a direct access file.

Now here are my solutions for how to open a direct access file using a variable. First of all I'll give you the quick and dirty way. I think that this will only work if the file is on disk drive 1. I have not found a way to open a file on drive 1 using a variable and the standard OPEN command.

HELPFUL HINTS

The first method works by using a variable to open a sequential file. Make sure that you pad the file name with spaces if it is less than 7 characters long, or it won't work properly and will be un-KILLable. AFTER the file is opened you can trick the APF system into thinking that the file is a direct access file by POKEing one location.
The pokes are:

```
IF FILE 0, POKE 41426,68
        1       41432,68
        2       41438,68
        3       41444,68
```

You can then read and write to the file using standard direct access commands. The file will still appear in the directory as a sequential file. It will have the same KILL problem as any other direct access file.

Now I'll explain the more difficult, but correct method. This method will allow using a variable for sequential or direct access file opening on either drive. The files will appear correctly in the directory.

My system is in machine language so it must be put into a REM statement. It uses about 58 bytes of memory and can be relocated with no changes (except for the CALL) if you have other machine language routines at &H407.

I have enclosed a printout from my assembler for anyone who wants to see what the routine does. This code can be entered directly using ROMMON or the APF monitor. I have also enclosed a BASIC program which sets up the routine and tests it. Line 0 is a REM statement which uses 58 A's to make room for the routine. Line 20 sets up where the routine will be placed in memory (in this case, &H407). Line 30 POKEs the code to memory and line 35 checks to see if there were any typing errors in the DATA lines.

Line 50 sets A$ up as seven spaces. This is necessary if the file name will be less than seven characters long. If this is not done, it is possible to get zeros in your file name, which makes the file un-KILLable without using DISKMOD. Line 90 is what you use instead of OPEN. CALL 41991 enters the routine which was poked into the REM statement. The comma separates the CALL from the file number and file name (A$). Another comma must be used to separate the file type and drive number from the file number and file name. If you are using drive 0 for the file, you do not need to enter a drive number. The file type must be included and must be "I" for a sequential file or "D" for a direct file.

HELPFUL HINTS

If you intend to use this routine in your program, you can have the BASIC program set up the routine in memory and then delete lines :0-310. The routine will be saved with your BASIC program as line -0-. Line -0- will list as garbage, of course.

I hope that this letter will clear up this APF problem for you.

Bill

Bill Lundquist
3739 Wentworth S.W.
Wyoming, Mi. 49509

```
0  REM  AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
10 REM : OPEN STATEMENT FIX BY BILL LUNDQUIST 3/4/1984
20 X=A(99): PRINT  TAB (5):" -- LOADING PATCH --": PRINT
30 READ A: IF A)999 THEN  POKE X,A:X=X+1:C0=C0=A: GOTO 30
35 IF C0(>5090 THEN  PRINT "YOU HAVE A DATA ERROR.  RE-CHECK": P
RINT "YOUR TYPING.": STOP
40 DIM P4(0): C0L0 0
50 PRINT :A4="          ": INPUT "ENTER DIRECT FILE NAME",A4
60 IF  LEN (A4))7 THEN C0T0
70 REM  USE THIS CALL INSTEAD OF -OPEN-
80 XA1    :=ST =0 =A#LE #A0=FILE NAME:D=FILE TYPE (DIRECT):2N
82 XA1    =B1VE 0
90 CALL A1991.0,A,D0
.0 PRINT : PRINT "DIRECT FILE HAS BEEN CREATED": PRINT "OR OPE
NED."
110 CLOSE 0: PRINT : DIR
120 DATA 49,49,254,160,37
130 DATA 201,42,160,6,132,3
140 DATA 183,16,285,8,255
150 DATA 60,37,189,107,241
160 DATA 254,160,37,6,230
170 DATA 0,193,73,39,7
180 DATA 193,63,39,3,126
190 DATA C0.117,8,255,160
200 DATA 37,109,110,6,126
210 DATA 36,19,0,999
```

```
                NAM  OPEN  FIX
        ****************************************
        * PATCH TO CORRECTLY HANDLE THE -B- AND*
        * -I- EXTENSION OF THE "OPEN" STATEMENT*
        * BY Bill Lundquist 3/4/1984           *
        ****************************************
(A407)         ORG  $A407       IN BASIC REM STATEMENT
A407 31        INS              REMOVE CALL RETURN ADDR. FROM STACK
A408 31        INS
```

HELPFUL HINTS

DIRECT ACCESS CONTINUED:BILL LINQUIST

```
A429 FE A025           LDX  #A025        GET BASIC BUFFER POINTER
A42C 08                INX               TO PASS OVER THE ","
A42D A6 00             LDA A 0,X
A42F 84 03             AND A #3
A431 B7 A1CD           STA A #A1CD       A-FILE # 0-3
A434 08                INX
A435 FF A025           STX  #A025        GET FILE NAME
A438 BD 6BF1           JSR  #6BF1
A43B FF A025           JSR  #A025
A43E 08                INX               TO PASS OVER THE ","
A41F B6 00             LDA A 0,X         S/B FILE TYPE -I- OR -D-
A421 C1 49             CMP B #'I
A423 27 07             BEQ  TYPOK
A425 C1 44             CMP B #'D
A427 27 03             BEQ  TYPOK
A429 7E 6E75           JMP  #6E75        "EXPRESSION ERROR"
A42C 08         TYPOK  INX               MOVE UP LINE POINTER
A42D BD A025           STX  #A025
A430 BD 6E86           JSR  #6E86        CALL APP OPEN COMMAND
A433 7E 800A           JMP  #800A        BACK TO BASIC

                00  ERROR(S) DETECTED


SYMBOL TABLE:
TYPOK    A42C
```

FROM DWIGHT WENSEL

"The R8-K can be used in tandem 1 to 4. Power off and follow the manual instructions. Place up to 4 R8-K's in the R8-1. Set switches to ON/ENABLE on ALL R8-K's. Turn power ON (Basic cartridge in place) and INITIALIZE. When cursor appears set all switches to OFF/DISABLE. Now each R8-K can be used separately by turning each one on individually using one at a time. Be sure to turn one OFF before you enable another. Data WILL NOT BE LOST while switch is OFF and writing to another. Data in R8-K WILL NOT be lost even through a RESET IF the switches are OFF! All data in the R8-K's share common addresses at (0000-DFFFF (hex. 49152-57343). I find it useful in machine language programming by keeping an original program in one location (R8-K) and using another to DEBUG and REVISE the program while keeping the original intact in the other cartridge."

FROM DWIGHT WENSEL

"I found a board that can be used in the B8-1 that is pre-drilled and copper clad on one side with holes at .1 off center. The board has a 25/50 socket and provides several busses and various arrangements for I.C.'s. The board is a RADIO SHACK CAT# 276-1647"

"I have also retrofitted the 'BASIC' printed circuit board with ZIP (Zero insertion force) chip sockets as BILL LINQUIST used. The IC-2 can be replaced with a MCM2532. The IC-1 can be replaced with a MCM6764. These are 8/4¼usable/programmable chips. The 2564 can be used with a 28 pin socket. Pins 1 & 28 are tied together internally and retrofitting requires a jumper to +5V. Pins 1 & 27 are then jumpered to GROUND. I did this on my first one before I found the MCM68764 (24pin)."

THANK YOU DWIGHT!

SHORT PROGRAMS

Here's a game written by VINCENT JOHNSON that tests your aim, judgement, and sharpshooting abilities! The object is to determine the location of an invisible target by information provided by the program, and to try to score a BULLSEYE with the amount of ammunition given. The program list continues on the next page. THANK YOU VINCENT!

```
10 GOTO 1000
20 MUSIC "4171* COLOR #0
21 PLOT C1,R1: RETURN
30 IF (C1-C2+R1)*(C1+(C1-T1)=(C2+(C2-T2))*R2)*(R2+R1)*R0 THEN FOR I=1 TO 10: MUSIC "4174* NEXT I: PRINT "OFF TARGET"
40 GOTO 110
50 CALL 17941: POKE 46963,0: POKE 40961,0
55 RETURN
100 REM BEGIN PLAY
110 SHAPE #15
120 B = ABS (C-C1): ARS OR-R1): REM DISTANCE FROM TARGET
130 POKE 40960,0: POKE 40961,0
140 PRINT "DISTANCE "; B; " CHOOSE LEFT "; "T"; R2
150 CALL 60: PLOT C1,R1
160 MUSIC "1 1"; I=0: KEY= (1)
170 IF Q0=*H* THEN GOSUB C0+R2+R1-1: GOTO 160
180 IF Q0=*P* THEN GOSUB C0*R1+R1+1: GOTO 160
190 IF Q0=*U* THEN GOSUB C0*(C1-R1): GOTO 160
200 IF Q0=*N* THEN GOSUB C0*(C1+C1-1): GOTO 160
210 IF Q0=*F* THEN MUSIC "171" IF C1=C THEN IF R1=R THEN GOTO 2000: REM A HIT
220 IF Q0=*F* THEN T=T-1: IF T<1 THEN GOTO 2300: REM NO SHOTS LEFT
230 IF Q0=*F* THEN GOTO 240
240 GOTO 160
400 POKE 34578,83: COLOR #0
410 MUSIC "1 1" INPUT "WANT THE INSTRUCTIONS (Y/N)*; Q0
500 IF Q0=*Y* THEN GOSUB 60: COLOR #000
540 GOSUB 60: GOTO 1000
1560 C= INT( RND (0)*20)+1: REM BULLSEYE COLUMN 1-20
1660 R= INT( RND (0)*14)+1: REM BULLSEYE ROW 5-14
1670 C1= INT( RND (0)*20)+1: REM RANDOM ROW COLUMN
1680 R1= INT( RND (0)*14)+1: REM GUN ROW
1890 GOTO 130
2000 REM EXPLODE ON
2010 GOSUB 40: CALL 17924: CALL 17926
2020 C=25: R=71: I=I+1
2030 COLOR #9: SHAPE #15
2040 FOR I=1 TO CH2 STEP 5
2050 FOR J=0 TO R+4 STEP 5
2060 PLOT I+J: NEXT J: NEXT
2070 C=C-1+R-1: I+J+I+N+J
2080 COLOR #11: IF C>T THEN GOTO 1100
2100 GOTO 2040
2150 POKE 40960,0: POKE 40961,0
2300 PRINT "BULLSEYE! OFF CENTER SPC (24)+"BULLSEYE*
2350 FOR I=0 TO 1500: NEXT I: GUN READ : PAUSE
2400 GOSUB 40: PRINT "PLAY AGAIN (CL KEY)*
```


BLIND TARGET

SHORT PROGRAMS

```
2160 QA= KEY$ (1): IF QA="P" THEN T=16: GOSUB 601 GOTO 9000
2150 IF QA="" THEN  GOTO 2140
2160 STOP
2500 REM  SOLUTION
2510 GOSUB 604: PRINT "SHAME ON YOU!": PRINT
2520 PRINT "NOW I'LL SHOW YOU WHERE YOU"
2530 PRINT "AIMED, AND WHERE THE BULLSEYE"
2540 PRINT "WAS,": FOR I=0 TO 500: NEXT
2550 GOSUB 601 GOTO 9000
3000 REM  SPC (10)"BLIND TARGET": PRINT
3010 PRINT "YOU'LL BE LOOKING AT AN UNUSUAL"
3020 PRINT "TARGET, (PAINTED BY A GUEST.)"
3030 PRINT "IT HAS A BULLSEYE, BUT YOU CAN'T"
3040 PRINT "SEE IT, NOR IS IT NECESSARY.": PRINT
3050 PRINT "IN THE CENTER,": PRINT
3060 PRINT "THE NEARER YOU WILL SEE IS WHERE"
3070 PRINT "YOU ARE AIMING THE GUN."
3080 PRINT "THE LEGEND ACROSS THE TOP SHOWS"
3090 PRINT "HOW FAR AWAY FROM THE BULLSEYE"
3100 PRINT "YOUR AIM IS, AND HOW MANY SHOTS"
3110 PRINT "YOU HAVE LEFT,": INPUT "MORE (RETURN KEY)";QA
3120 GOSUB 604: PRINT "MOVES DON'T COST YOU ANY BULLETS"
3130 PRINT "BUT YOU HAVE 'FIRE' TO SEE HOW"
3140 PRINT "CLOSE YOU ARE, HITTING THE"
3150 PRINT "BULLSEYE IN LESS THAN 10 SHOTS"
3160 PRINT "IS A SNAP,": PRINT
3170 PRINT "USE THE RIGHT JOYSTICK AND THE"
3180 PRINT "'FIRE' BUTTON,": PRINT
3190 INPUT "READY TO START (RETURN KEY)";QA: GOSUB 601 RETURN
9000 REM  DRAW TARGET
9010 PRINT "DISTANCE= "Y$:"SHOTS LEFT= "IT
9015 COLOR +3: SHAPE +15
9020 PLOT 5:24 PLOT 5:15: PLOT 26:24 PLOT 26:15
9030 SHAPE +142: I= FOR C2=6 TO 15
9040 PLOT C2:R2: NEXT
9050 C2=25 FOR C2= TO 25
9060 PLOT C2:R2: NEXT
9070 C2=15 FOR R2=6 TO 14
9080 PLOT C2:R2: NEXT
9090 C2=16 FOR R2=1 TO 14
9095 PLOT C2:R2: NEXT
9100 COLOR +0: REM  NO SOLUTION  YET
9120 SHAPE +15: COLOR +3
9130 PLOT C:R
9150 COLOR +4: PLOT C1:R1
9160 POKE 40960:2: POKE 40961:0
9170 INPUT "OKAY (RETURN KEY)";QA
9190 GOTO 2330
```

BLIND TARGET

PRODUCT REVIEW

AMWAY SA-13 by CARTHAN ASSOCIATES

This program is offered by CARTHAN AS-SOCIATES—Washington, D.C. and is a special-ized program written solely for AMWAY distributors wherever they may be.

Because of it's specialized nature, I will attempt to cover the main points of the program in this review and ask that anyone who may have further questions to write CARTHAN ASSOCIATES for full de-tails. Their address appears in the CLAS-SIFIED SECTION.

Basically the program (186) is made up of DATA statements that pertain to AMWAY products, their individual prices and stock numbers. The statements are in alpha-betical order starting with HOME CARE PRODUCTS and continuing on with many other products (Over 200).

The program operation is simple and air-eaightforward. You are prompted to input the total amount of sales of such an individual product by it's appropriate pro-duct number. Initially you are asked for the data and order number. You must have the printer ready with your AMWAY totals sheet in place. After asking you for the order number, a HEADING containing your name and address along with your AMWAY number and sponsor is outputted to the printer. The quantity of product number are asked for and the cursor disappears for a while as it searches for the number. Once found,the information is pulled from the DATA statement and displayed by it's product description on the screen. If you want it printed, a carriage return is needed.

The information that is sent to the printer is a tally of each sale per product number and an overall summary of you sales pertain-ing to your purchase volume and order date. The program is formatted expressly for the AMWAY totals sheet.
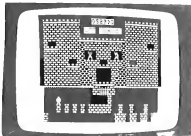
The large amount of DATA statements con-tained in this program limits other poss-abilities that it could have such as storage of data for future reference, menu driven control of the program itself, or comparisons, etc.. However, if people involved with AMWAY products find this program easy to operate and affecting the way in which it is structured, they are sure to make good use of it.

GRATEKULL by George Karabin

If you appreciate excellent computer gra-phics, fast moving challenging games that include high variety and an equal amount of difficulty, read on!

One of the newest games from GEORGE KARABIN called GRATEKULL includes these charact-eristics and is rated TOPS by everyone we've heard from.

One outstanding feature of this game is that it really gives you a feeling of accomplishment while playing it. The play is very smooth, the action is believable, and the enjoyment of simply playing the game seems to outweigh the necessity of obtaining a HIGH SCORE. Here are some photos of the game in action along with a brief overview.
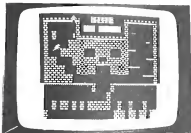

The player is given 100 seconds on the clock initially. You control (joystick) a little brawny guy called HEMAN and at-tempt to maneuver him to different areas outside and within the castle GRAYSKULL. HEMAN runs east and west, jumps up, leaps NORTHEAST, NORTHWEST, bravely draws his sword to deflect bombs dropped by those inside the castle, or can be positioned with his sword drawn facing left or right to ward off the onslaught of rolling rocks. The object is to get to as many areas within the castle as possible and return to the starting point within a given length of time. Each time HEMAN reaches the HIGH-EST point in each tower, additional points are awarded, and if he returns to his starting point before time runs out, ad-ditional time is given to attempt either a different tower or a more difficult

PRODUCT REVIEW

one. Once both towers have been entered and climbed a BATTLE ROYALE (PHASE 3) takes place in the MAIN HALL of the castle. At this point HEMAN encounters the ultimate confrontation with two characters called SKELATOR and BRASTHAR. These guys stand atop the castle throughout the game and throw bombs and rocks at HEMAN. If HEMAN draws his sword and deflects these weapons, more points are added to the score. If he is hit by one, a 2 second time deduction results.

Once the battle has ended, HEMAN finds himself back at the starting point ready to advance another round to increase the score even higher!

Each round becomes more difficult. For instance, HEMAN must LEAP over some pilings to get to the castle. Initially the pilings are fairly easy to get to with a healthy leap. Later, the pilings have sizable gaps in them and you must be more careful with your leaps that you don't fall through. If this happens, HEMAN must be restituted so that he can try another leap. This takes time to do and time is fleeting in this game. Also, notice the perches in the right tower. After a round is completed, the perches contain gaps so that it is more difficult to accomplish a successful leap without falling through a gap and ending up at the base of the tower.

This game provides hours of fun for all ages. It is exciting, variety filled, and challenging. The character movement is very realistic and the actual game play is extremely interesting and FUN!

DATA SHOP

As mentioned last month, there are various types of modems available today, but the most common type is the DIRECT CONNECT modems. These modems are considered as BELL STANDARD 103 for 300 BAUD or BELL STANDARD 212 for 1200 BAUD modems. You may find a large assortment of options and/or features in these types such as AUTO ANSWER, CHARACTER BUFFERED, AUTO ORIGINATE, REVERSE CHANNEL, and so on, but if they are classified as 103 or 212 they will work with the IM-1.

Any modem that is classified as PRIVATE LINE is a DEDICATED LINE modem and requires a dedicated physical line in order to pass HIGH SPEED DATA. These types are normally found where extremely large amounts of data must be exchanged between two or more systems in a short amount of time. The cost of the dedicated facility, the modem itself, and the requirements normally prohibit their use in the home computer environment. This article will relate to the direct connect dial-up type of modem only.

BAUD RATE

The term BAUD RATE is used to describe the SPEED in which data is transmitted. At 300 BAUD for example data is transmitted at approximately 30 characters per second (CPS) which is 1/10th the BAUD RATE. The 300 BAUD modem is rated to pass data at a MAXIMUM of 30 CPS but will operate at a slower rate such as 110 and 150 BAUD. It is very important to remember that the BAUD RATE that is set on your DC-232 or SI-232 must match the BAUD RATE setting of the equipment at the distant end but must not exceed the maximum BAUD RATING of the modems used on either end. Also, if a printer is used at the same time as the modem is used, the BAUD RATE on the printer must be set to match that of the IM-1. In other words, set ALL DEVICES to the same BAUD RATE SETTING.

CONNECTION

Most modems come with a standard RS232 cable and connector, however, the buyer must be aware of the CONNECTOR TYPE (male or female) and the cable requirements before ordering. There are several cable types available with various pin placements and lead configurations. The connector required to interface a modem to the DC-232 or SI-232 is a standard RS232 type (DB-25) MALE 25 pin connector. Most modems cables do not come with ALL 25 pins wired because normally only a few are required to make the modem fully operational. Below is a list of the leads normally required when connecting a modem to the IM-1.

SI-232 or DC-232		MODEM CABLE	
PIN		PIN	
1 FRAME GROUND		1 FRAME GROUND	
2 TRANSMIT DATA		2 RECEIVE DATA	
3 RECEIVE DATA		3 TRANSMIT DATA	
4 REQUEST TO SEND		4 REQUEST TO SEND	
5 CLEAR TO SEND		5 CLEAR TO SEND	
6 DATA SET READY		6 DATA SET READY	
7 SIGNAL GROUND		7 SIGNAL GROUND	
8 CARRIER DETECT		8 CARRIER DETECT	
20 DATA TERMINAL READY		20 DATA TERMINAL READY	

The modem you choose may vary from this pin arrangement and may not correspond to this chart, however, the SI-232/DC-232 pin designations are correct and your modem cable should be re-pinned to match. Modems such as the ANCHOR MARK VII and VOLKSMODEM require cable lead changes (re-pin) before they will work with the IM-1.

NEXT MONTH... a look at the PRINT-1 and PRINT-2 functions and some samples of data exchange between TWO IM-1's using these commands.

SHORT PROGRAMS

★ ★ ★

```
1  REM  BERNARD E. COUTURE
2  REM  WATERVILLE REGIONAL VOCATIONAL CENTER
3  REM  WATERVILLE MAINE ##0##LETTER LOCATER#
4  DIM A#(2), D#(42)
5  CALL ITEM#(0+A$)=1)
6  N= INT ( RND (0)*26+1)
7  IF A=60 THEN 10
8  IF A=69 THEN 50
9  ON N GOTO 12,13,14,15,16,17,18,19,20
10 A=A+50: ON N GOTO 21,22,23,24,25,26,27,28,29
11 A=A+50: ON N GOTO 30,31,32,33,34,35,36,37
12 L$="A": GOTO 100
13 L$="B": GOTO 100
14 L$="C": GOTO 100
15 L$="D": GOTO 100
16 L$="E": GOTO 100
17 L$="F": GOTO 100
18 L$="G": GOTO 100
19 L$="H": GOTO 100
20 L$="I": GOTO 100
21 L$="J": GOTO 100
22 L$="K": GOTO 100
23 L$="L": GOTO 100
24 L$="M": GOTO 100
25 L$="N": GOTO 100
26 L$="O": GOTO 100
27 L$="P": GOTO 100
28 L$="Q": GOTO 100
29 L$="R": GOTO 100
30 L$="S": GOTO 100
31 L$="T": GOTO 100
32 L$="U": GOTO 100
33 L$="V": GOTO 100
34 L$="W": GOTO 100
35 L$="X": GOTO 100
36 L$="Y": GOTO 100
37 L$="Z": GOTO 100
100 PRINT L$
105 A= KEY$ (0)
110 IF A<=L$ THEN 600
115 IF X>15 THEN 500
120 GOTO 100
500 CALL 17##&: PRINT "YOU PRESSED THE WRONG KEY!!": PRINT "YOU ARE GOING TO HAVE TO 5": PRINT "BETTER!!"
510 FOR I2=1 TO 2: MUSIC " 7 2"#: NEXT I: GOTO 600
600 CALL 17##&: PRINT "YOU LUCKED OUT THIS TIME": PRINT "BUT I'LL GET YOU NEXT TIME!": MUSIC "50 5#": GOTO 5
700 CALL ITEM &: PRINT "GOOD LATE TO TRY!!": PRINT "YOUR TIME HAS EXPIRED!": MUSIC "4### 4## 4# ### ## 2### 1## 2###"
710 PRINT "PRESS RETURN TO CONTINUE": INPUT X$: GOTO 5
```

LETTER
LOCATER

THE ARCADE

We stopped last month in mid-stream after
typing in a short BASIC program, just
before examining the data contained in
the program. Refer to the MAY issue and
type the program in again. After entering
the program type CALL28672 to enter the
monitor mode.

Brio will begin this issue explaining
the data beginning at A407.

Trying typing in D4407. This is one of the
first addresses in your USER REN. The
41's are all the A's in your first line of
BASIC. Remember line 10 and what it
contained. Now hit carriage return and
try typing in MA407. With this command
you only display 1 line at a time but
can also change the data AT the address
you are looking at by simply typing in
the new data and depressing the slash
(/) key. We use the N command to enter
our programs and you can also use it to
look at data only. Try typing in MA407
again. Now if you hit the / key without
entering any data, the next address (A408)
will be displayed along with its data.
Keep hitting the / key until you get a
different number than 41 as your data.
The next numbers are 0D which means that
this is the end of a line in BASIC. The
next 2 locations are 00 and 20 which is
the NEXT BASIC LINE NUMBER in your program.
The next few addresses contain the ML
code (no the said) REM STATEMENT which
are followed by a line of 72's. These
are all the 8's that we put into BASIC
line 20. Continue to / or list and you
should come to all zero's which is the
end of your BASIC program. If you now
hit shift N or the comma, the lines will be
displayed going backwards. This is very
handy when going back just a few lines.
The third command is the G command or
GOTO. With this command, the Monitor
will go to any address you specify after
typing the G. GA407 would immediately
jump to that address and start running
from that point. We can use the G command
to get us out of the monitor mode and
back into BASIC by typing in GB494. This
puts us back into BASIC where you will
again see your normal cursor.

One important thing to stress here is

that it is crucial to remember never to
use the REK number 0D in your MACHINE
LANGUAGE PROGRAM. This is an interpreted
function that means END OF LINE MARKER
and cannot be used in ML operations. This
symbol is used in BASIC as explained pre-
viously.

There are ways to program around using
HEX 0D or 13 in decimal. We'll learn later
that in HIRES, our screen memory is from
HEX address 0000 to 0180. There are two
screen addresses which contain 0D and
they are 0000 and 0105. Since we will
want to use the whole screen for our game,
we will have to learn how to program
around using 0D in our ML program.

Let's do a comparison in speed of ML versus
BASIC. We'll start with a BASIC program
that moves a block across the screen.
In LOW RES graphics we can use the alpha
characters or the graphics characters.
Every graphic character has a HEX code
that can be directly put into the screen
memory or on your TV. The code for a white
block is decimal 15 or HEX 0F. If we poke
15 to the screen, you would see the block.
The low res screen is from decimal address
512 to 1023 or HEX 0200 to 03FF. Address
512 is in the upper left corner of the
screen and there are 32 blocks across
to 543 in the upper right corner. Address
544 would be below 512 and it continues
for a total of 16 rows down. The whole
screen is made up of 16 rows of 32 squares.
If we poke 79[,]15 we see a white block
in the middle of the screen. To erase
it you can poke 79[,] with 128 (BLACK) or
with a 32 (GREEN).

Type in the following BASIC routine and
run it. You will see a white square moving
across the screen and from top to bottom.

```
5CALL17046
10POKE+51 2tol023
20POKEI,15
30FORX=0TO50:NEXT
3OCALL61 32
40NEXTI
3OPRINT"DONE":PRINT:PRINT:STOP   (Now RUN)
```

Notice the speed that the shape moves
across the screen. Next month we'll do
this program in ML. CSAVE this program
for next month's instruction. See you
then!

The following information was submitted by GARY SHILTON - Bloomington, Illinois. The list contains important data pertaining to USER DATA Addresses, INTERNAL CONTROL FUNCTIONS, SERIAL INTERFACE FUNCTIONS, TAPE INTERFACE, VARIABLE LABEL LIST, and CURSOR LOCATIONS. This is a PARTIAL listing of some of the IM-1 system operations contained within the code ROM. The information was gathered from the PROGRAMMING TECHNICAL ASSISTANCE MANUAL and is provided as a quick reference. Thank you Gary for your work in compiling this list!

Overview of On-Board Addresses

Contains all APPLE*MARK.

A0060	4A060	MSB of CURSOR Location. First byte saved on Entry
A0061	4A061	LSB of CURSOR position. (LEAST Significant Byte)
A0067	4A007	MSB of LOW memory address/location to SAVE to TAPE.
A0068	4A008	LSB of LOW memory address to SAVE to TAPE.
A0069	4A009	MSB of HIGH memory address to SAVE to TAPE.
A006A	4A00A	LSB of HIGH memory address to SAVE to TAPE.
A00FE	4A0FE	Address to MOVE TO. (Used in assembly routines)
A0FBA	4A0FA	LSB (to move blocks of memory to new adr-locations.)
A1BFF	4A0FB	LSB of MEM FROM. (n/a above routine.)
A1BFE	4A1FE	LSB 3 with # bytes to be moved-call BREAK+0TP041.
A1BFF	4A1FF	MSB of NEXT FREE adr. to STORE the CONTENTS of a VAR.
A1BFE	4A1FE	LSB of NEXT FREE adr. to STORE the CONTENTS of a VAR.
A121B	4A12B	(will contain a VARIABLE TABLE LIST.)
...VARIABLE..		Each LABEL=9 bytes. First 2 bytes are for search=ASCII.
.ABL..	4A7C5	The remaining 7 bytes use info. in another variable
		is of the SIMPLE= SINGLE or DOUBLE subscript form; the
		two-byte locations and if numeric or string variable.
A1AFE	4A0FE	2nd variable LABEL.
A1AFE	4A0FE	3rd variable LABEL.
A13DB	4A13C	END of VARIABLE TABLE-LIST. (26 var. # 9 + 234 bytes)
A14FE	4A14F	MSB counter (FLAG) indicates END of SYSTEM MEMORY.
A14FF	4A15F	LSB counter (FLAG) indicates END of SYSTEM MEMORY.
A14FB	4A14B	
A14FC	4A14C	
A14FD	4A15D	
A14FE	4A15E	
A14C2	4A15C	FLAG-POINTER to indicate which 512-byte block of memory to save to TAPE. If + 0= bias screen memory to saved.
		Division of INPUT/OUTPUT BUFFER...see below.
A14DB	4A14D	(top of I/O BUFFER. Like! 127 bytes from 40760)
A14C0	4A14C	High-order-FLAGS NEXT FREE ADDL. TO STORE A PROGRAM STEP
A14D5	4A14L	LSB pointer-FLAGS NEXT FREE ADDL. TO STORE A PROGRAM STEP
A14FE	4A14F	FLOWING ADDR. OF FLAGS FREE USER MEMORY space.
		(reverse contains)
40651	4069FF	END OF MEMORY for a 9k system. (USER MEM.=7+166 bytes)
57343	4GFFF	END OF MEMORY for a 16k system. (USER MEM.=15+357 bytes)

SPACE PANIC

It was announced in the February issue of the newsletter as Krazy Kong -- a version of Donkey Kong. Well, as I worked on the game the name Space Panic came into mind. Similar to the Donkey Kong screen you have to reach the top of a building by climbing up ladders. Also, you have to try to jump over the space men coming after you. This is the first and only version of a Donkey Kong style game available for the APF computer. The regular price is $7.50 but from now until July 1, 1984 it is $1.00 off!

SALE PRICE $6.50
+ $1.00 shipping

ALLIEN DEFENDER II

First you played it at the arcades, then you were able to play it on your APF home computer, and now there is a new better version of this popular game. You have to try to shoot down or avoid the coming alliens, if you can. The original Allien Defender has sold more than all the other games I have made and since the reviews were so good I decided to revise it. If you have already purchased the original then send me the tape with $1.00 and I will place it on the back. As with Space Panic it is on sale until July 1, 1984 for $1.00 off.

JUST LOOK AT WHAT YOU GET:

* High resolution graphics with many sounds!
* Front loading Screen!
* Detailed instructions!

*MAJOR DIFFERENCE FROM ORIGINAL: One of the alliens locks onto you and you cannot avoid it.

SALE PRICE $7.00
+ $1.00 shipping

STAR CASTLE

You've seen it in the arcades and now you can have it for your APF home computer. A high speed game which requires an extremely lucky player. You are in command of a small space ship with a mission to destroy the star castle. Fire as much as you like, but good luck getting past the protective wall and look out for the castle's laser gun. This 8K program comes with detailed instructions.

For only $7.50
(including shipping)

(REVISED) APF BASIC LANGUAGE REFERENCE MANUAL

Do you ever get tired of trying to translate the mistakes APF Electronics made when writing the manual that came with the computer? Well, now there is an alternative, the new (Revised) APF Basic Language Reference Manual. The most complete manual available for the APF home computer. It includes everything the original manual contained (excluding the mistakes) except for high resolution graphics. Don't worry, the high resolution graphics manual will be ready in one to two months. Also, there are three special reference charts in the back. Such as a screen map chart and a shade/color chart. Please note that this is not a teaching manual, but merely for reference!

FOR ONLY $5.50
(including shipping)